Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024

Introduction:

In this lab, we're building a cooling system called a swamp cooler using an Arduino microcontroller and sensors. Evaporation coolers offer an energy-efficient alternative to traditional air conditioners, particularly in dry, hot climates. The system operates by drawing in outside air through a water-soaked pad, where evaporation occurs, cooling and humidifying the air. However, it's crucial to note that this cooling mechanism is ineffective in humid environments.

Description of Functionality:

This setup has different parts that either work with code or are controlled by the user directly. The Arduino controls things like the LCD, temperature and humidity sensor, clock, LED lights, water sensor, and motor. There's another power source for a motor and some buttons. Here's how it works: The sensor constantly checks the temperature and humidity. If the water level is good and there are no issues, the screen shows these values. The user can use a knob to change where the air vent points. If it's too hot, the fan turns on and a blue light comes on (Running State). If it's cool enough, the fan stops and a green light comes on (Idle State). The time shows up on the screen too. If the water level is low, a message shows up on the LCD and a red light comes on (Error State). During any of these states, the user can press the on/off button to send the system into the Disabled State (Yellow light on).The user can stop and restart the system using two buttons. One button switches between off, and on. The other button checks the water level and puts the system into an idle state from the error state if sufficient water has been added.

Description of Code:

In the setup, the UART is initialized along with the adc, button interrupts, lcd, rtc, and the ddr ports. In our main loop, we used a function to control the stepper motor vents. Then we had our main switch statement for the various states of the system. Using an enumerated value, we were able to distinguish what state the system is in.

States:

In the Idle state, we turned on the green light, made sure the fan was off and checked temp and water in case we needed to change states. In the Error state, we turned the red light on, fan off, and printed out an error message on the lcd. In the Running state, we turned the blue light on, fan on, and checked water and temp to see if we needed to change states. In the Disabled state, we turned the yellow light on and fan off. We then checked to see if our state had changed from the previous state and if it did, we would print the timestamp and state to the serial monitor.

Checks:

In our check temp we used the dht11 library to check temperature and humidity levels, outputting them to the lcd. If the temp was above 20 degrees Celsius, it would send the system into idle or remain in idle, if not, it would continue as normal or send it into the running state.

Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024

In our check water level, we used the adc lab to analog read from the water level sensor. When the water level was below 100, it would send the system into an error state and send a message to the lcd.

Functions:

We made several functions to assist such as printTime which would use the rtc library to print the current time to the serial monitor. We also made some helper functions for our UART communication so that we could print out strings and multiple characters at a time. We had a my_delay function from a previous lab to delay how often it would output the temperature onto the lcd screen.

ISR's:

We had two interrupt functions which allowed the on/off button and the reset button to function. The on/off button would toggle the system from disabled to running and vice versa.

Constraints:

Water:

- If the water reached below 100, the system would go into an error state.

Temperature:

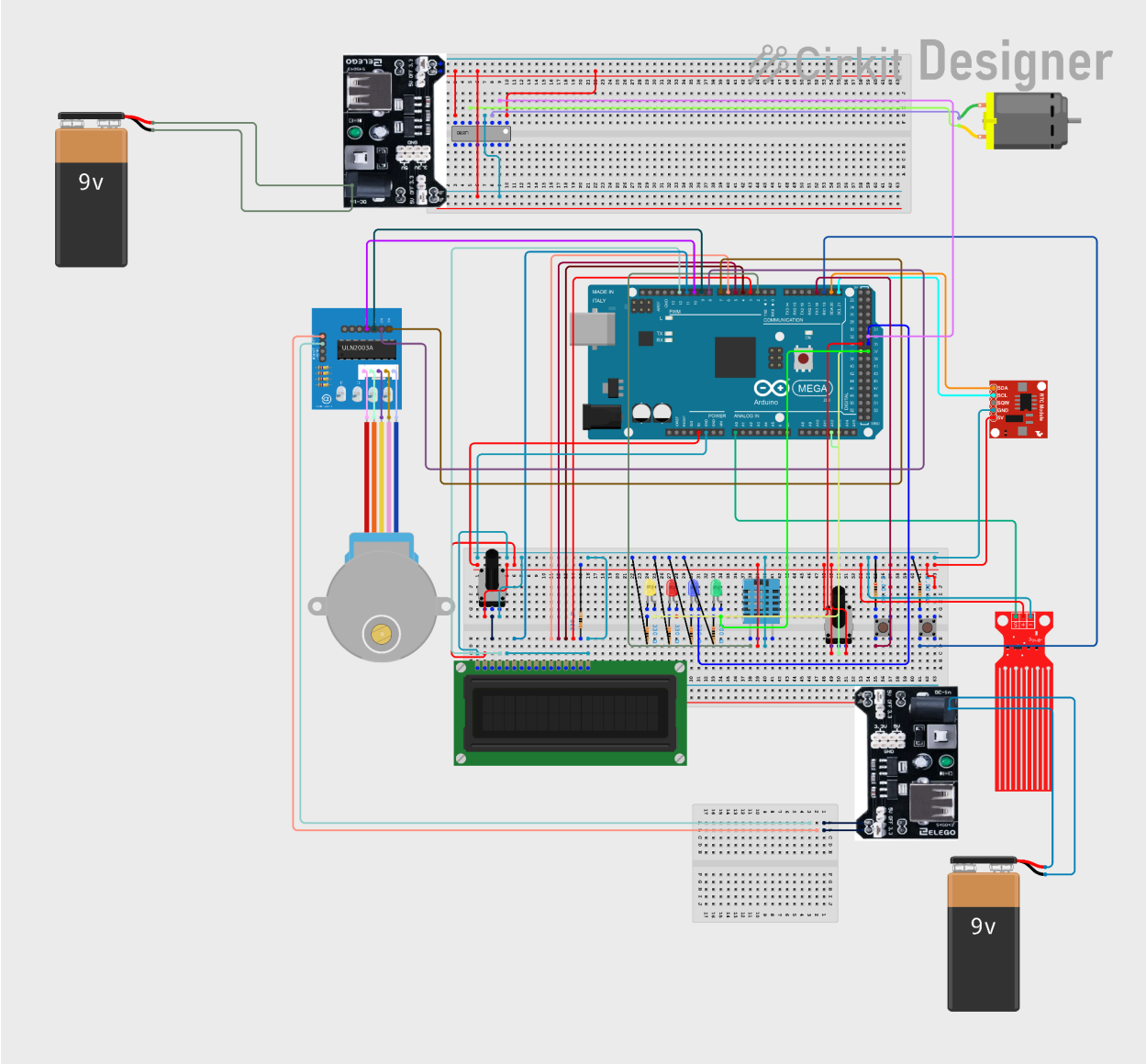
- If the temperature reached above 20 degrees celsius the system would turn the system to running and below 20 degrees goes to Idle state.

Power Sources:

- We used two 5-9V power sources in our project.

Schematics:

Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024



Ardiuno Mega 2560 Rev3: https://docs.arduino.cc/hardware/mega-2560/

ULN2003 Stepper Motor Driver:
https://www.electronicoscaldas.com/datasheet/ULN2003A-PCB.pdf

LCD Display: https://docs.arduino.cc/learn/electronics/lcd-displays/

Stepper Motor: https://docs.arduino.cc/learn/electronics/stepper-motors/

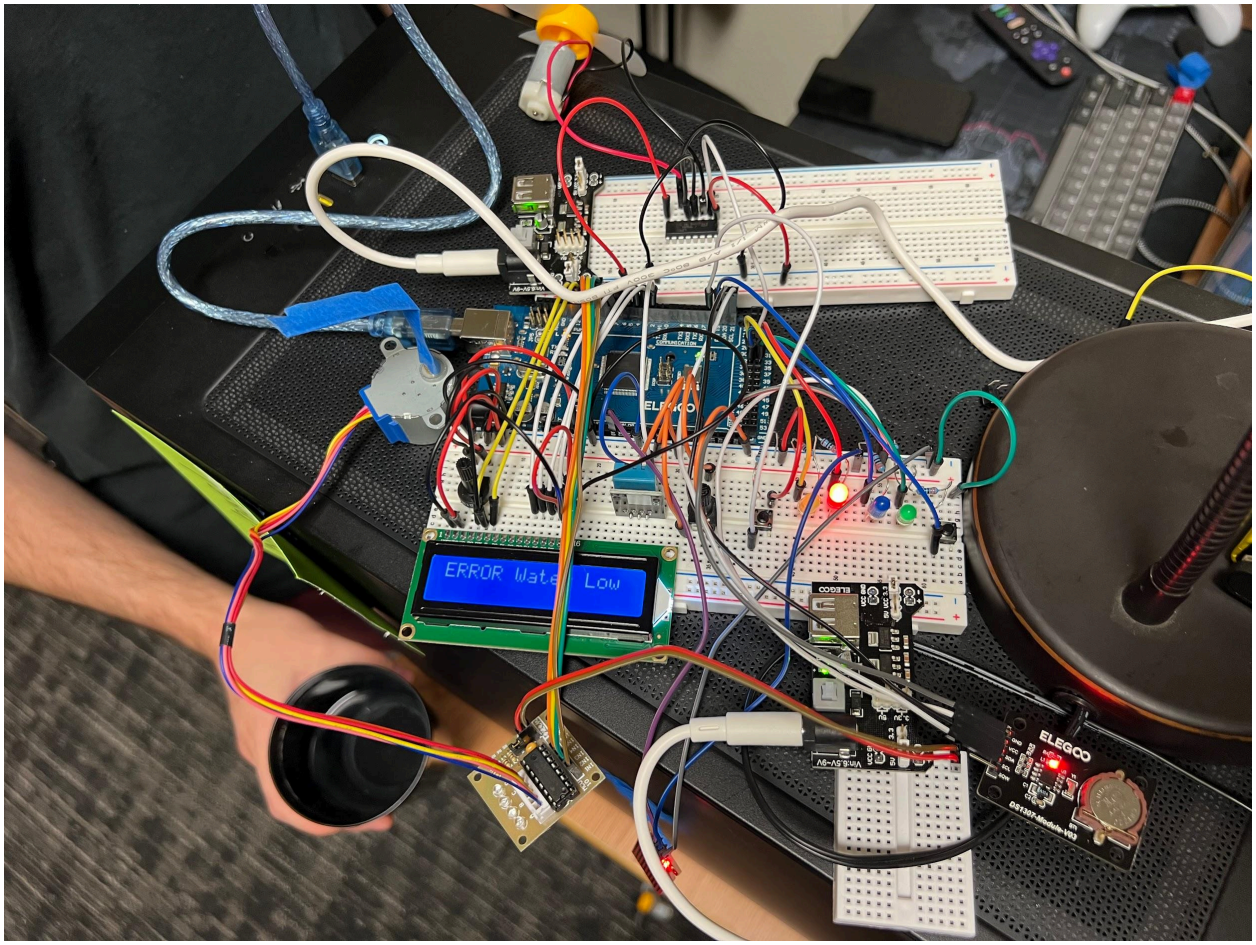
Water Sensor: https://arduinogetstarted.com/tutorials/arduino-water-sensor

Temperature and Humidity Sensor:
https://www.circuitbasics.com/wp-content/uploads/2015/11/DHT11-Datasheet.pdf

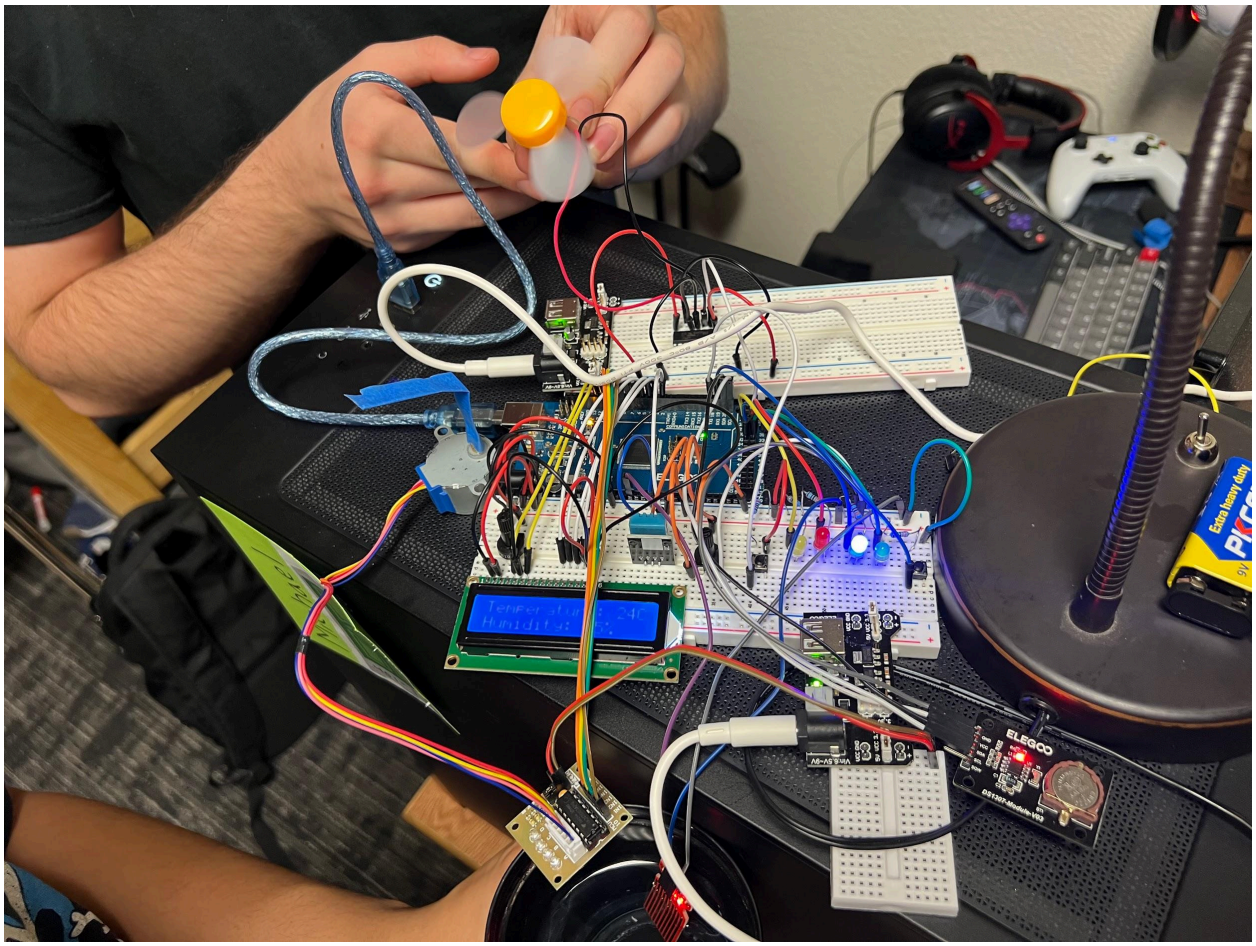
Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024

RTC Module:
https://www.analog.com/media/en/technical-documentation/data-sheets/DS1307.pdf

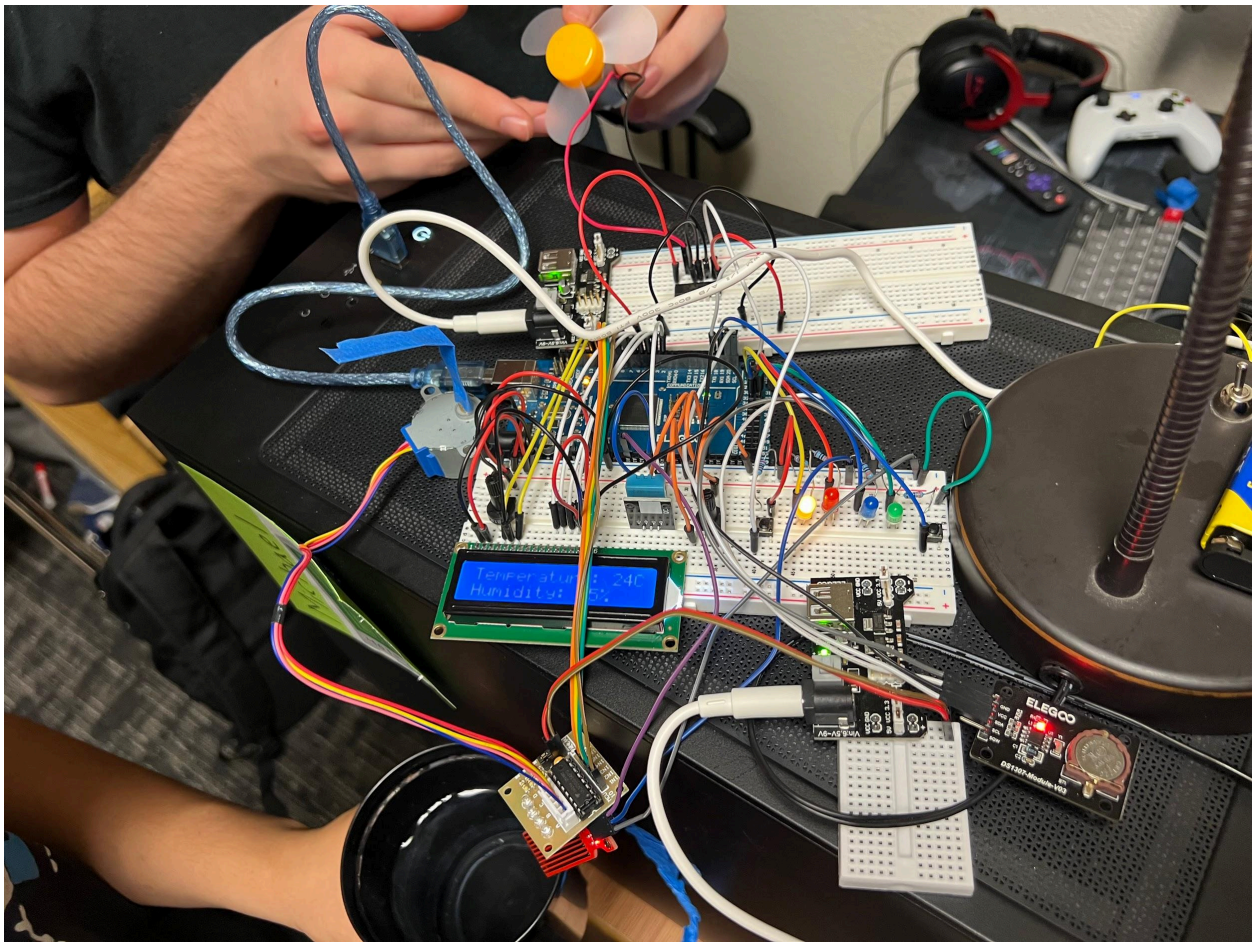
Circuit of Cooling System:



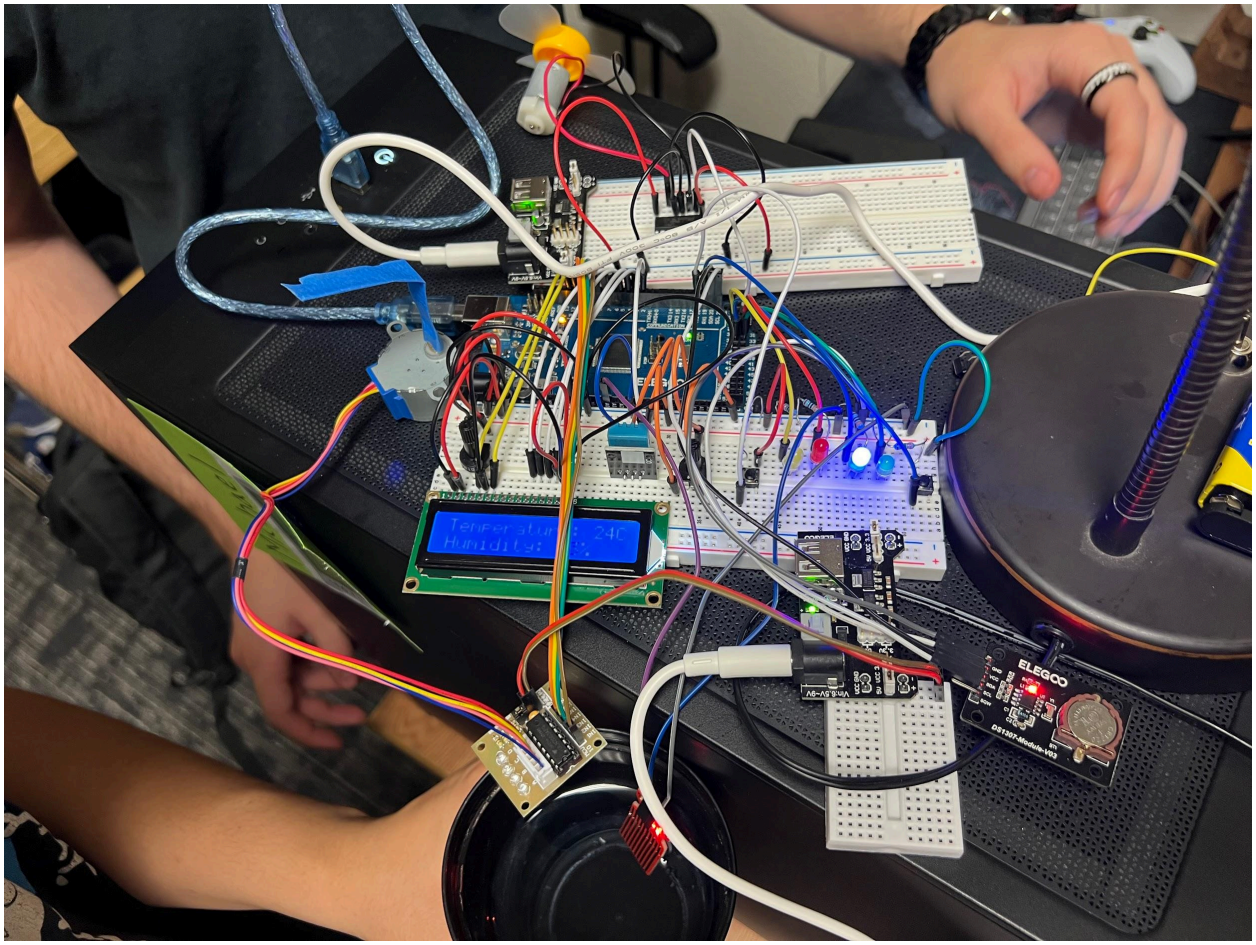
Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024



Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024



Michael Pedote, Thompson Thai, and Z He
CPE Final Lab
May 12, 2024



GitHub Repository:

https://github.com/1204-Pedote-Michael/CPE_301_FinalProject

Video:

https://youtu.be/jvg9EvIY4T8?feature=shared